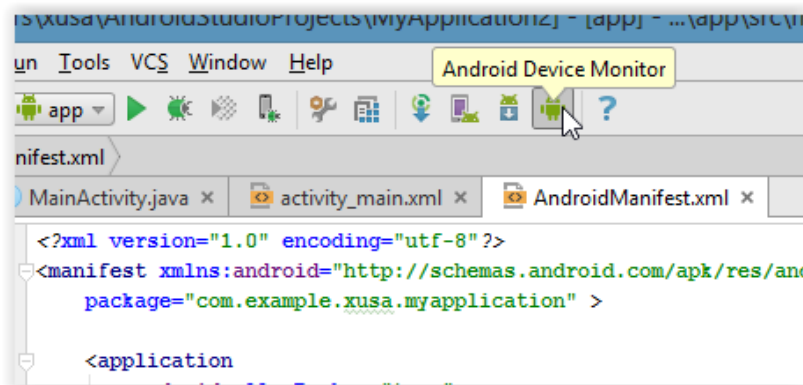


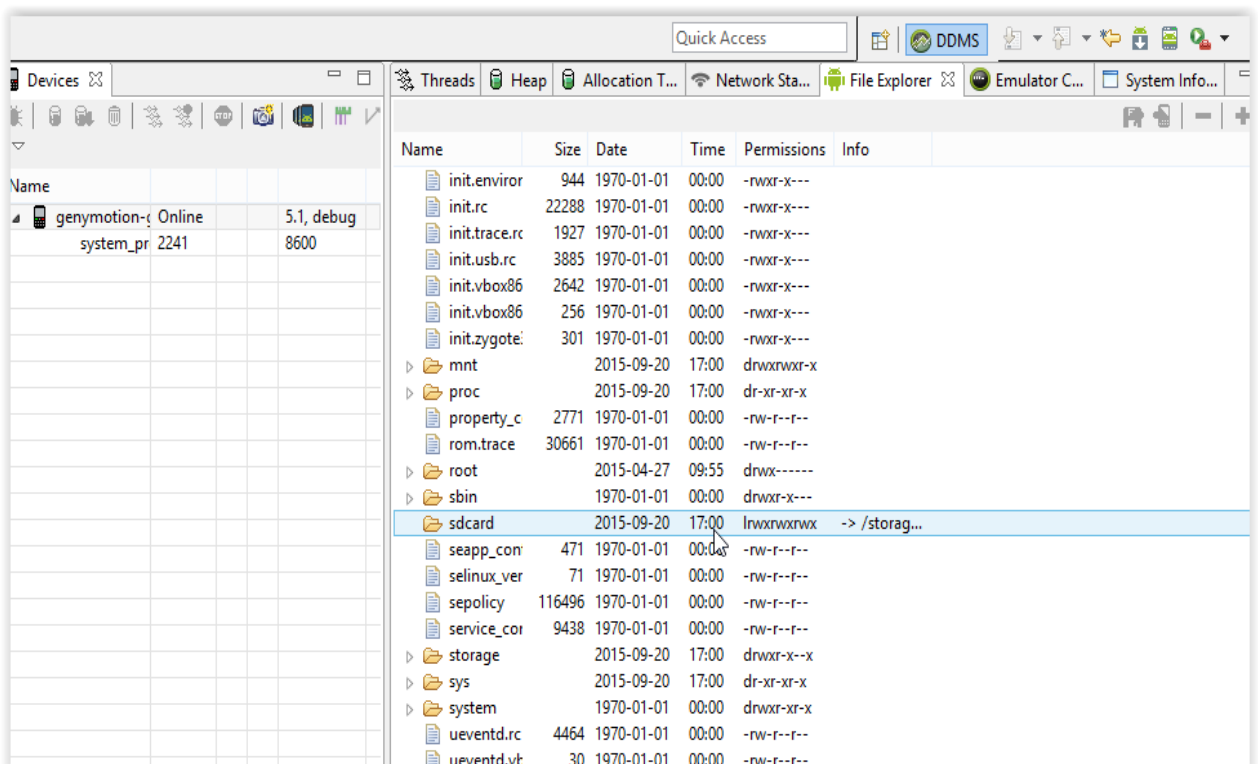
Realizar una aplicación que contenga tres botones. El primer botón llamado Abrir tiene que lanzar una canción mp3 que se encuentra en la tarjeta sd del dispositivo. El segundo botón nos permitirá ejecutar el navegador con la URL [www.iesdoctorbalmis.com](http://www.iesdoctorbalmis.com) y el tercer botón permitirá abrir una aplicación externa.

## PASO I. Acceso a la tarjeta SD

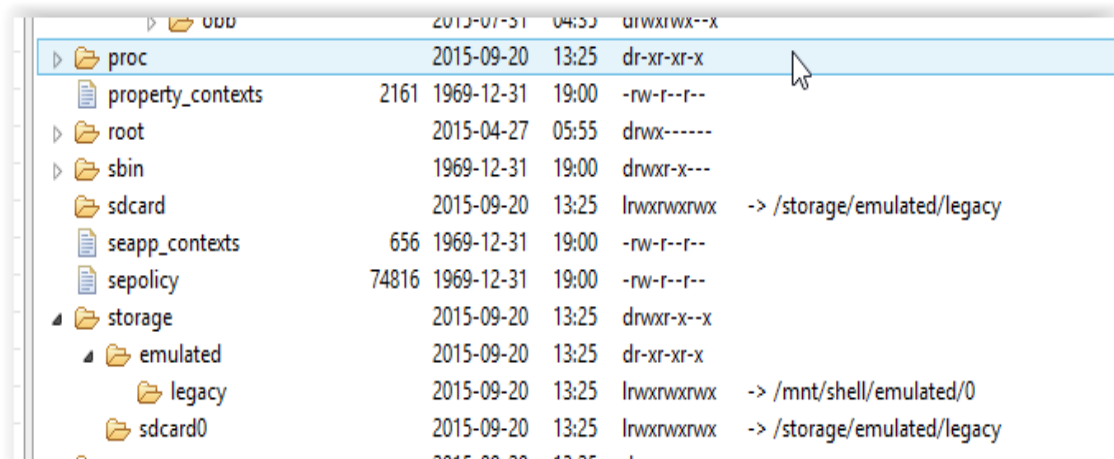
Después de crear el proyecto debemos acceder al Android Device Monitor, tenemos un icono de acceso en la barra de herramientas.



Ahora podemos ver la ventana donde administramos los archivos contenidos en la tarjeta SD del emulador.

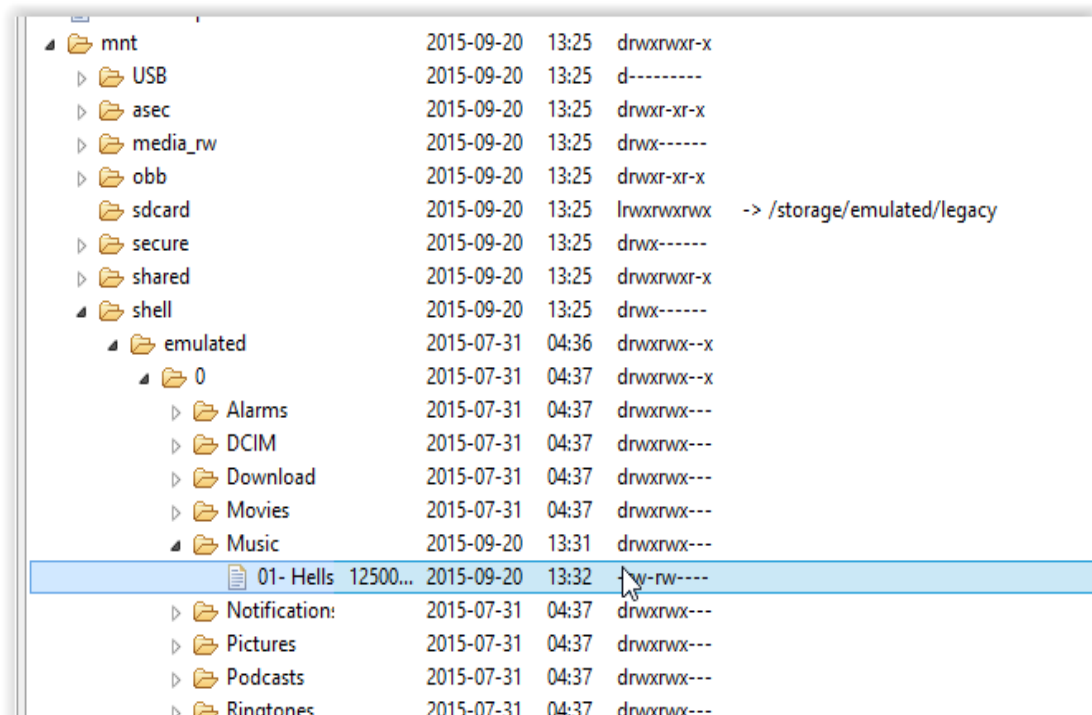


En la carpeta sdcard subiremos el archivo mp3. Pero para ello debemos fijarnos en las rutas reales a las que nos redirige cada una de las carpetas.



Nombre	Tamaño	Fecha	Hora	Permisos	Ruta real
proc		2015-09-20	13:25	dr-xr-xr-x	
property_contexts	2161	1969-12-31	19:00	-rw-r--r--	
root		2015-04-27	05:55	drwx-----	
sbin		1969-12-31	19:00	drwxr-x---	
sdcard		2015-09-20	13:25	lrwxrwxrwx	-> /storage/emulated/legacy
seapp_contexts	656	1969-12-31	19:00	-rw-r--r--	
sepolicy	74816	1969-12-31	19:00	-rw-r--r--	
storage		2015-09-20	13:25	drwxr-x--x	
emulated		2015-09-20	13:25	dr-xr-xr-x	
legacy		2015-09-20	13:25	lrwxrwxrwx	-> /mnt/shell/emulated/0
sdcard0		2015-09-20	13:25	lrwxrwxrwx	-> /storage/emulated/legacy

Si nos fijamos sdcard nos manda a la carpeta legacy de storage y a su vez esta nos manda al mnt..... Pues en la carpeta 0 es donde tendremos el directorio music y será donde arrastraremos el fichero de música que queráis.



Nombre	Tamaño	Fecha	Hora	Permisos	Ruta real
mnt		2015-09-20	13:25	drwxrwxr-x	
USB		2015-09-20	13:25	d-----	
asec		2015-09-20	13:25	drwxr-xr-x	
media_rw		2015-09-20	13:25	drwx-----	
obb		2015-09-20	13:25	drwxr-xr-x	
sdcard		2015-09-20	13:25	lrwxrwxrwx	-> /storage/emulated/legacy
secure		2015-09-20	13:25	drwx-----	
shared		2015-09-20	13:25	drwxrwxr-x	
shell		2015-09-20	13:25	drwx-----	
emulated		2015-07-31	04:36	drwxrwx--x	
0		2015-07-31	04:37	drwxrwx--x	
Alarms		2015-07-31	04:37	drwxrwx---	
DCIM		2015-07-31	04:37	drwxrwx---	
Download		2015-07-31	04:37	drwxrwx---	
Movies		2015-07-31	04:37	drwxrwx---	
Music		2015-09-20	13:31	drwxrwx---	
01- Hells 12500...		2015-09-20	13:32	-w-rw----	
Notification:		2015-07-31	04:37	drwxrwx---	
Pictures		2015-07-31	04:37	drwxrwx---	
Podcasts		2015-07-31	04:37	drwxrwx---	
Rinotones		2015-07-31	04:37	drwxrwx---	



## PASO2. Lógica Botón Abrir

Veamos el código y pasemos a las explicaciones:

```
3 import android.os.Bundle;
4 import android.os.Environment;
5 import android.app.Activity;
6 import android.content.Intent;
7 import android.net.Uri;
8 import android.view.View;
9 import android.view.Menu;
10 import android.widget.Button;
11 import android.media.MediaPlayer;
12 import android.net.Uri;
13
14
15 public class MainActivity extends Activity {
16     private Button bAbrir;
17     private Button bNavegar;
18
19     @Override
20     public void onCreate(Bundle savedInstanceState) {
21
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main);
24
25         bAbrir=(Button)findViewById(R.id.cmdOpen);
26         bAbrir.setOnClickListener(new View.OnClickListener() {
27
28             @Override
29             public void onClick(View v) {
30                 Uri datos = Uri.parse(Environment.getExternalStorageDirectory().getPath() + "/Music/01- Hells Bells.mp3");
31                 MediaPlayer mp=MediaPlayer.create(MainActivity.this, datos);
32                 mp.start();
33             }
34         });
35     }
36 }
```

La clase MediaPlayer nos permite reproducir sonidos (de diversos formatos) desde cualquier fuente disponible: Internet, fichero local o un recurso de nuestra propia aplicación.

Como siempre tenemos la definición del botón y su asociación con el botón del layout, esto se hace en la línea 25 (con *findViewById*).

A continuación definimos el *listener* asociado al botón y la redefinición del método *onClick*.

Creamos un objeto de la clase Uri llamando al método *parse* donde indicamos el path y nombre del archivo a recuperar (línea 30).

Creamos el objeto de la clase MediaPlayer pasando ahora la referencia del objeto de la clase Uri, línea 31.

Iniciamos la reproducción, línea 32.

OTRA FORMA:

Es posible también implementarlo a través de un intent. Veamos cómo hacerlo:

```
File datos = new File("/sdcard/music/hellsbells.MP3");
intento=new Intent(Intent.ACTION_VIEW);//Te abre aplicaciones de
sonido
intento.setDataAndType(Uri.fromFile(datos), "audio/mp3");
startActivity(intento);
```



### PASO3. Lógica Botón Navegar

Veamos el código y pasemos a las explicaciones:

```
37         bNavegar=(Button)findViewById(R.id.cmdBrowser);
38         bNavegar.setOnClickListener(new View.OnClickListener() {
39
40             @Override
41             public void onClick(View v) {
42                 Intent intent1;
43                 intent1=new Intent();
44                 intent1.setAction(Intent.ACTION_VIEW);//acción externa, abrirá el navegador
45                 intent1.setData(Uri.parse("http://www.iesdoctorbalmis.com"));
46                 startActivity(intent1);
47             }
48         });
49     }
50 }
```

### PASO4. Lógica Botón AbrirAppExterna

Veamos el código y pasemos a las explicaciones:

```
51
52         bAbrirApp=(Button)findViewById(R.id.cmdOpen2);
53         bAbrirApp.setOnClickListener(new View.OnClickListener() {
54
55             @Override
56             public void onClick(View v) {
57                 Intent intent = new Intent(Intent.ACTION_MAIN);
58                 intent.setComponent(new ComponentName("hola.usuario", "hola.usuario.MainActivity"));
59                 startActivity(intent);
60             }
61         });
62     }
63 }
64 }
```

OTRA FORMA: a través de `getPackageManager`, que permite buscar entre todos los paquetes que tenemos instalados el que se indica con `getLaunchIntentForPackage()`.

```
bAbrirApp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = getPackageManager().getLaunchIntentForPackage("hola.usuario");
        startActivity(intent);
    }
});
```

El nombre del paquete debe ser único, por lo que para asegurarnos que cuando se suba a GooglePlay se siga la nomenclatura:

com.nombrepaginawebdesarrollador.nombreakcion

NOTA:

**URI** son las siglas en inglés de **uniform resource identifier** (en español identificador uniforme de recursos), que sirve para identificar recursos en Internet.

Dicho identificador tiene un formato estándar y su propósito es permitir interacción con recursos disponibles en Internet -o en alguna red de cómputo-, como por ejemplo páginas, servicios, imágenes, vídeos, etc.

